Reward: https://codefights.com/img/coins_new.png **1000**

Find the least significant one bit and return it.

Example: 12 (decimal) = 1100 (binary) LSOne(12) = 0100 (binary) = 4 (decimal)

**Input (N)** → integer :

**Output** → integer :

The least significant 1 bit of N

<https://codefights.com/feed/xeJPaytfc8P4T5ubN>

**----Mi solución ACEPTADA-----**

#include <iostream>

#include <vector>

#include <stdio.h>

#include <algorithm>

#include <math.h>

int LSOne(int N)  {

        std::vector<bool > bin;

*/\* convierto el decimal a Binario*

*hasta el primer uno mas a la derecha correspondiente al num en binraio\*/*

        while(N % 2 == 0) {

                bin.push\_back(N % 2);

                N /= 2;

        }

        bin.push\_back(N % 2); */\*agrego el primer uno \*/*

*/\* al nummero binario invertido lo transformo a decimal \*/*

        int dec = 0;

        for(int i = 0; i < bin.size(); i++) {

*/\* si es true = 1, false = 0\*/*

                int dig = ( bin[i] ) ? 1 : 0;

                dec +=  dig \* pow(2, i );

        }

        return dec;

}

int main() {

        int res = LSOne(12);

        printf("%d ", res);

        return 0;

}

---------VERSION COMBINADA-------

int LSOne(int N)

{

    int dec = 0;

    int i = 0;

    while (N % 2 == 0)

    {

        dec += (N % 2)\* (int)(pow(2, i));

        N /= 2;

        i++;

    }

    dec += (N % 2) \* (int)(pow(2, i));

    return dec;

}